

### **REMARKS**

Claims 1-64 are currently pending in the subject application and are presently under consideration. Claims 1, 38 and 58- 63 are amended, solely to clarify their statutory class. Claim 64 is newly added. Support for the claim amendments and new claim 64 may be found, for example, in the specification at paragraphs [0031], [0032] and [0135]-[152].

Favorable reconsideration of the subject patent application is respectfully requested in view of the comments and amendments herein.

#### **I. Examiner Interview**

Applicant's representative thanks Examiner Dave for the courtesies extended in the telephonic interview held March 5, 2009, the substance of which is incorporated into these remarks. In the interview, the features of the independent claims were discussed in view of Rehof et al. (US 2003/0204570). No agreement was reached.

#### **II. Rejection of Claims 1-44 and 58-63 Under 35 U.S.C. §101**

Claims 1-44, and 58-63 stand rejected under 35 U.S.C. §101 as allegedly being directed to non-statutory subject matter. Withdrawal of the rejection is respectfully requested in view of the amendments herein which clearly place the subject claims within the bounds of statutory subject matter in accordance with 35 U.S.C. §101.

#### **III. Rejection of Claims 1-17 and 19-63 Under 35 U.S.C. §102(e)**

Claims 1-17 and 19-63 stand rejected under 35 U.S.C. §102(e) as being anticipated by Rehof et al. (US 2003/0204570).

Withdrawal of this rejection is respectfully requested for at least the following reasons. Rehof et al. does not disclose or suggest each and every feature recited in the subject claims.

Applicants' claimed subject matter generally relates to a novel programming language extension for expressing asynchronous computations. Advantageously, the claimed subject matter can enable a programmer to write interface contracts that prescribe the ordering and overlap of a collection of asynchronous functions. The language can support asynchronous programming by way of additional programming constructs and a behavioral type system that statically detects common errors. Further, the language can support an alternative concurrent

programming model based on futures, asynchronous function calls, and joins on futures, which are building blocks for asynchronous systems.

In particular, independent claim 1 recites ...*a conformance checking component that checks that at least one of the client interface and the service interface conform to the contract defined therebetween*... . The Office Action incorrectly asserts that paragraph [0009] of Rehof et al. discloses the noted features (Office Action, p. 4, 3<sup>rd</sup> par.). Specifically, the Office Action states “behavioral analysis checks whether the implementation (of the message passing program module) conforms to an intended set of type processes.”

However, the cited section (let alone the entire reference) fails to point out anything resembling a contract as recited in claim 1. A contract requires at least two participants, but Rehof et al.’s par. [0009] relates only to a single “implementation.” In more detail, par. [0009] describes a “system for evaluating properties of a message-passing module.” To this end, the system “checks whether the implementation conforms to an intended set of type processes.” The type processes “specify certain message-passing actions that the implementation is to perform as well as communication channels restricted for the actions.” Thus, the operations described in Rehof et al. are one-sided; that is, they relate only to the particular implementation of the message passing program module.

Accordingly, the Office Action fails to identify in Rehof et al. anything corresponding to a contract defined between a client interface and a service interface (let alone checking for conformance with the contract), as recited in claim 1.

Regarding independent claim 38, Rehof et al. fails to disclose “a conformance checking component that checks that the client interface obeys the behavioral relationship, and the service interface properly implements the behavioral relationship,” as recited in the subject claim. The Office Action again relies on Rehof et al. at par. [0009] for the noted features. However, as discussed supra, Rehof et al. describes one-sided operations relating only to a particular implementation of a message passing program module, and is silent *regarding checking conformance to a relationship between a client interface and a service interface*.

As to independent claim 45, along similar lines, Rehof et al. does not disclose “checking to ensure that the client interface obeys the behavioral relationship and the service interface properly implements the behavioral relationship,” as recited.

Similarly, with respect to independent claim 54, Rehof et al. fails to disclose “enforcing the behavioral relationship between the client interface and the service interface,” as recited.

Likewise, as to independent claim 58, Rehof et al. is silent regarding “an interface model from the contract state; and a conformance checking component that checks that the client model obeys the interface model, and the service model properly implements the contract model”.

As to independent claim 63, along lines discussed previously, Rehof et al. fails to disclose “means for checking to ensure that the client interface obeys the behavioral relationship and the service interface properly implements the behavioral relationship,” as recited.

Furthermore, numerous dependent claims recite novel features (above that of respective base claims) that are neither disclosed nor suggested by Rehof et al. For example, claim 2 recites “the contract defines an order in which clients of the service interface are invoked.” The Office Action cites par. [0009] of Rehof et al. for the noted features. However, par. [0009] makes no mention of defining an order as claimed, and in fact, the Office Action only refers generally to “various behavioral properties of the message passing program module” (Office Action, p. 4, par. [0004]). This is insufficient to support the asserted rejection.

Claim 3 recites “the checking component checks that the clients handle both normal and exceptional results.” The Office Action cites par. [0032] and par. [0050] of Rehof et al. for the noted features. The Office Action alleges a correspondence between the claimed “exceptional results” and “paragraph 0050, various type annotations that specify intended message passing actions for the program module” (Office Action, p. 5, 1<sup>st</sup> par.). However, an “intended” message passing action, as referred to in the Office Action, is clearly not “exceptional” as recited in claim 3.

Claim 5 recites “the checking by the checking component is modular wherein the contract is specified on an interface boundary of the client interface and the service interface and, each of the client interface and the service interface is checked independently, and each method is checked separately.” The Office Action cites Rehof et al. at par. [0027] and par. [0032] for the noted features. However, since, as noted previously, Rehof et al. is silent regarding a contract as claimed, it follows that Rehof et al. is still further deficient concerning operations involving a contract “specified on an interface boundary” as recited in claim 5.

As a still further example, claim 8 recites “a construct in the form of an asynchronous call that returns a future-like handle to an eventual result of the call.” The Office Action cites Rehof

et al. at par. [0010] for the noted features. However, par. [0010] merely describes “type annotations,” which in no way correspond to returning a “future-like handle to an eventual result of the call,” as recited in the subject claim.

Claim 10 recites “a construct that states the behavioral contract on the service interface.” As noted previously, Rehof et al. is silent regarding a behavioral contract as recited in claim 10.

As a still further example, claim 11 recites “a construct that is a transaction directive that delineates the scope of conversations by clients.” The Office Action cites Rehof et al. at par. [0010] for the noted features. However, as noted previously, par. [0010] describes type annotations and is silent concerning a transaction directive as claimed.

Claim 12 recites “a construct that is a tracked type qualifier used by the checking component.” The Office Action cites Rehof et al. at pars. [0010] and [0011]. However, pars. [0010] and [0011] describes type annotations and are silent regarding the features of this claim.

Claim 19 recites “the value is an object in one of three states that include an undefined state, normal state, and exceptional state.” The Office Action cites Rehof et al. at par. [0010]. However, as noted previously, par. [0010] relates to type annotations and makes no mention of a value taking on three states.

As a still further example, claim 20 recites “the contract is a source-level contract that facilitates statically detecting asynchronous message passing errors.” Since, as noted previously, Rehof et al. is silent regarding a contract as claimed, it follows that Rehof et al. is still further deficient concerning operations involving a “source-level contract”.

Similarly, claim 21 further defines the contract set forth in claim 1, reciting “the contract expresses stateful protocols that govern interactions between a client associated with the client interface and a service associated with the service interface.” Consequently, Rehof et al. is still further deficient with respect to claim 21 for the additional features it recites over claim 1.

Claim 22 likewise further defines the contract set forth in claim 1, reciting “the service interface is associated with a service contract whose language models the effects of multiple clients that concurrently access the service interface.” Consequently, Rehof et al. is still further deficient with respect to claim 22 for the additional features it recites over claim 1.

Along similar lines, claim 23 further defines the contract set forth in claim 1, reciting “the checking component checks that services and clients obey the contract associated with the

service interface.” Consequently, Rehof et al. is still further deficient with respect to claim 23 for the additional features it recites over claim 1.

Claim 24 recites “the checking component uses a transaction directive to delimit a scope of an instance of a concurrent interaction between a client and a set of service instances.” The Office Action cites Rehof et al. at par. [0010] for the noted features. However, as discussed previously, par. [0010] merely relates generally to type annotations and is silent as to any of the features recited in the subject claim.

As a still further example, claim 25 further defines the transaction directive set forth in claim 24, reciting “the transaction directive specifies a correlation variable that uniquely identifies the instance of the concurrent interaction between the client and the set of service instances.” Consequently, Rehof et al. is still further deficient with respect to claim 25 for the additional features it recites over claim 24.

Claim 30 recites “the checking component further comprises an extraction component that uses a region to model relevant data, which relevant data is that data which is directly relevant to the contract state.” The Office Action cites Rehof et al. at pars. [0027], [0031] and [0032] for the noted features. However, par. [0027] merely describes a distributed client-server computing environment, while pars. [0031] and [0032] relate to type annotations and behavioral “abstraction” of a program module. There is no mention in Rehof et al. of using a region to model relevant data as recited in claim 30, in particular data relevant to a contract state, since Rehof et al. is silent concerning a contract as discussed previously.

Moreover, claim 32 further defines the region set forth in claim 30, reciting, “the region is partial such that only relevant data is directly assigned to a region type.” Consequently, Rehof et al. is still further deficient with respect to claim 32 for the additional features it recites over claim 30.

Yet in another example, claim 34 recites “the checking component further comprises an extraction component that utilizes region-and-effect analysis to extract a model.” The Office Action cites Rehof et al. at pars. [0076]-[0077] for the noted features. However, pars. [0076]-[0077] relate to detecting type annotations in source code, and make no mention of using region-and-effect analysis to extract a model, as recited in claim 34.

In view of the foregoing, independent claims 1, 38, 45, 54, 58 and 63 (and claims 2-17, 19-37, 40-44, 46-53, 55-57, and 59-62 which respectively depend therefrom) are allowable over

Rehof et al., and this rejection should be withdrawn.

**IV. Rejection of Claim 18 Under 35 U.S.C. §103(a)**

Claims 18 stands rejected under 35 U.S.C. §103(a) as being unpatentable over Rehof et al. Withdrawal of this rejection is respectfully requested for at least the following reasons. Claim 18 depends from independent claim 1, and is therefore allowable over Rehof et al. for at least the reasons noted above regarding claim 1.

Moreover, claim 18 recites “the result returned is explicitly retrieved through a join statement.” The Office Action contends “Rehof et al. discloses combining the data from multiple actions and efficiently process[ing] large quantities of data [taking a number of actions with respect to other events], so it would be obvious to use a Join statement because it was a commonly used statement at the time of the invention” (Office Action, p. 23, 2<sup>nd</sup> par.).

In response, it is noted that the Supreme Court, in *KSR*, quoting *In re Kahn*, stated that “[R]ejections on obviousness cannot be sustained by mere conclusory statements; instead, there must be some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness.” In view of the foregoing, it is respectfully submitted that the Office Action fails to provide a rational underpinning for the rejection of claim 18. Claim 18 recites retrieving a result through a join statement; the Office Action argues that such a feature is suggested simply by the general description in Rehof et al. of “combining ... data” and “process[ing] large quantities of data.” It is respectfully submitted that such general operations, common to most data processing, in no way suggest the features explicitly recited in claim 18.

Accordingly, claim 18 is allowable over Rehof et al.. Withdrawal of the rejection is therefore respectfully.

**V. New Claim**

New claim 64 is clearly allowable over the art of record. None of the art of record discloses or suggests a system that facilitates asynchronous programming, comprising one or more computing devices configured with a contract component that facilitates stating a behavioral contract on an asynchronous service interface between a client interface and a service interface, and a conformance checking component that checks that at least one of the client interface and the service interface conform to the contract defined therebetween, further

comprising a procedure, which procedure is a method that is called at least one of synchronously and asynchronously, the asynchronous call executes concurrently with its caller by completing immediately, while the method that is invoked executes, the asynchronous call completes by returning a value, the value is used explicitly to synchronize the asynchronous call with the value, the value is a first-class type that can be at least one of stored, passed as an argument, and returned as a result, the result returned is explicitly retrieved through a join statement, and the first class type is introduced by one of a local declaration or a field member declaration, as required by the combined features of claim 1, claims 13-18 dependent thereon, and new claim 64 dependent on claim 18.

**CONCLUSION**

The present application is believed to be in condition for allowance in view of the above comments and amendments. A prompt action to such end is earnestly solicited.

In the event any fees are due in connection with this document, the Commissioner is authorized to charge those fees to Deposit Account No. 50-1063 [MSFTP447US].

Should the Examiner believe a telephone interview would be helpful to expedite favorable prosecution, the Examiner is invited to contact applicants' undersigned representative at the telephone number below.

Respectfully submitted,

AMIN, TUROCY & CALVIN, LLP

/William E. Curry/

William E. Curry

Reg. No. 43,572

AMIN, TUROCY & CALVIN, LLP  
57<sup>TH</sup> Floor, Key Tower  
127 Public Square  
Cleveland, Ohio 44114  
Telephone (216) 696-8730  
Facsimile (216) 696-8731